SECTION III—REMARKS

Applicants respectfully request reconsideration of the above referenced patent application for the following reasons:

Interview with the Examiner:

On Wednesday, May 05, 2010, Applicants' representative, Spencer K. Hunter, a law clerk working under the discretion of Gregory D. Caldwell, the undersigned attorney of record, discussed the present case in detail with Examiner James Rutten.

Although a formal agreement was not reached with respect to particular claim limitations to place the case in immediate condition for allowance, Examiner Rutten did indicate that proposed amendments which recite certain novel aspects taught within Applicants' specification were moving the case "in the right direction." Examiner Rutten stated that further consideration on his part was necessary before coming to a determination with respect to the proposed claim limitations, and invited Applicants to submit "well reasoned arguments," upon which he could rely in making his determination with respect to the proposed limitations over the presently relied upon art of record.

With respect to the particular limitations discussed, Applicants' representative argued that U.S. Patent No. 6,662,359 to Berry et al. (hereinafter "Berry") fails to disclose a mechanism which implements a "**user-configurable level of granularity** specified via a Graphical User Interface presented at an end-user device" by "modifying bytecode of only **a subset of a plurality methods from which the application is composed**," where the "subset of the plurality of methods:"

> ... **provides the <u>user-configurable level of granularity</u>** by providing the functionality for tracing the program flow of the application through only the subset of the plurality of methods specified via the tracing and debugging operations injected into the subset of the plurality of methods specified

Applicants placed special emphasis on the fact that Berry is silent with respect to the mechanism for specifying **how** a subset of methods belonging to an application is selected for modification. For example, despite Berry's disclosure of an "inclusion/exclusion list" at column 7, line 7, Berry provides no discussion whatsoever how such an "inclusion/exclusion list" is utilized with respect to the generation of a "**subset** of a plurality of methods from which [an] application is composed."

Conversely, Applicants proposed clarifying amendments and discussed such amendments with Examiner Rutten that expressly recites how a "**subset** of the plurality of methods" is determined, in more detail than Berry can reasonably be interpreted to disclose.

Specifically, Applicants recite in the proposed claim (which is presented herein for entry into the record and formal consideration), a method for implementing the "**user-configurable level of granularity**" by:

> **presenting to the end-user device** via the Graphical User Interface, **options for modifying the application's bytecode** by injecting tracing and debugging operations into the application's bytecode ...
>
> **<u>modifying</u> bytecode of only a subset of a plurality methods** from which the application is composed, the subset of the plurality of methods selected from two or more class files in two or more archive files composing the application's bytecode as **specified via the Graphical User Interface** presented to the end-user device ... .

Thus, Applicants proposed limitations which explicitly tie the ability to determine such a "subset" based on "options for modifying the application's bytecode" to the "end-user device."

Applicants' representative noted during the interview that such a mechanism is helpful to an end-user to, for example, prevent an end-user from being bombarded with tracing and debugging output from a mechanism, such as the one which is proposed by Berry, which appears to modify all the code/methods of a particular application, or at the very least, modify code/methods of a particular application without providing an end-user device or an end-user of such a device the ability to specify a "**user-configurable level of granularity**."

Because Berry does not provide for such a mechanism, it is impossible for a user of Berry's proposed mechanism to control the "level of granularity" as Applicants proposed to recite in amended claims, and thus, Applicants' representative argued that Berry should be overcome.

The clarifying amendments proposed during the Examiner's interview are set forth above as amendments for formal entry into the record. Applicants address the proposed amendments and the presently relied upon references, including Berry, in additional detail below with respect to the outstanding rejections pending in the case at bar. Applicants further address additional detail now recited by the amended claims, which was not discussed during the Examiner's interview.

Although a formal agreement for allowability was not reached during the interview, Applicants thank Examiner Rutten for granting the interview, and request the Examiner to please consider the more detailed arguments for allowability in view of Applicants more detailed and clarifying limitations presented in the amended claims. Applicants now address the rejections set forth in the present Office Action.

Claim Rejections - 35 U.S.C. § 103

The Office Action rejected claims 40-54 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,662,359 to Berry et al. (hereinafter "Berry") and U.S. Patent No. 6,560,618 to Ims (hereinafter "Ims"). Applicants respectfully disagree. For example, independent claim 40 recites in pertinent part:

> A method for modifying an application to provide functionality for tracing a program flow of the application at a user-configurable level of granularity specified via a Graphical User Interface presented at an end-user device, the method comprising:
>
> ...
>
> presenting to the end-user device via the Graphical User Interface, **options for modifying the application's bytecode by injecting tracing and debugging operations into the application's bytecode at the *user-configurable level of granularity specified* via the Graphical User Interface**, wherein the application is composed of a plurality of archive files associated with two or more distinct tiers in a multi-tiered architecture, the archive files having respective class files, and the respective class files having respective methods, and wherein the options for modifying the application's bytecode includes:
>
> **modifying bytecode of only a subset of a plurality methods from which the application is composed**, the subset of the plurality of methods selected from two or more class files in two or more archive files composing the application's bytecode **as specified via the Graphical User Interface** presented to the end-user device, wherein the two or more archive files are **associated with two or more distinct tiers of the multi-tiered architecture**, and wherein the modified subset of the plurality of methods specified provides the user-configurable level of granularity by providing the functionality for tracing the program flow of the application through only the subset of the plurality of methods specified via the tracing and debugging operations injected into the subset of the plurality of methods specified ... .

Overview of the present invention:

So as to aid the Examiner in a more efficient examination of the present case and so as to expedite the issuance of a notice of allowance, Applicants provide the following overview of that which is taught by Applicants' specification. Applicants note that while this overview is provided to aid in a fuller understanding of the present invention, Applicants limit the scope of the claims presented only to the explicitly recited limitations of the claimed embodiments.

In the specification as originally filed, Applicants teach that a mechanism is provided that allows for a program flow of an application to be traced at varying levels of "precision." For example:

> **[0022]** A system and method are described for tracing program flow within an application. In one embodiment of the invention, options for modifying application bytecode **at a variety of different levels of precision are provided**. For example, bytecode may be modified at the application level, the package level, the class level and/or the method level.

Such a mechanism is directed toward overcoming the additional complexity which is introduced into a system that utilizes a multi-tiered architecture in which disparate archive files and source code is distributed among different tiers and in which a program flow does not follow a simple top-down path within a more conventional self contained "monolithic" program, in which all references and functionality is localized. The need thus arises to be able to trace a program flow through this more complex landscape, while at the same time, allowing the trace to only provide relevant information as directed by, for example, a user-device or a user. Hence Applicants' recitation of the term, a "**user-configurable level of granularity**" in the present claims. Refer to, for example, paragraph 21 from Applicants' specification which teaches:

> **[0021] Although the multi-tiered system illustrated in Figure 2b provides a more flexible and scalable architecture, it also results in significant additional complexity**. For example,

> monitoring, testing and/or debugging multiple clusters of presentation layer servers, business layer servers and databases, and the dependencies between them requires a significant amount of management overhead. As such, the ability to efficiently monitor, test and/or debug object-oriented, virtual-machine-based enterprise software, such as the software employed in a J2EE environment, is critical for efficient software development and/or implementation.

Thus, Applicants specification does not simply teach the ability to insert tracing logic into a file, but rather, Applicants teach a mechanism by which pertinent methods (e.g., units of functionality) dispersed throughout a more complex environment can be selectively modified so as to provide an end-user device with an appropriate amount of "precision" or render program-flow tracing information at a "**user-configurable level of granularity**." For example, refer again to Applicants' specification which teaches:

> [**0116**] … Thus, when the user-configurable plugin 820 is employed (as opposed to the application tracing plugin 810), the method invocation tree **does not include entries for *all* of the methods** of an application. Rather, it **only includes entries** for methods within the particular package or class, or the individual methods **selected by the end-user.**

As noted above, providing for tracing capabilities at a "user-configurable level of granularity" is not a simple matter of editing the source code of a more traditional "monolithic" type application having localized source code, rather, in the more complex **multi-tiered** environment, the methods which provide functionality to an application may be **within different packages within the architecture**, creating complexity when a tracing operation of an application. For example, refer again to Applicants' specification as follows:

> [**0115**] Thus, in contrast to the application tracing plugin 810 which causes the bytecode modifier 452 to modify all of the methods within a particular application, the user-configurable plugin 820 illustrated in Figure 8 provides a finer level of granularity for tracing program flow. **An "application" may be built from a *plurality of packages* (typically *.jar files in a Java**

> **environment); each package may be built from a plurality of classes (i.e., class files); and each class include a plurality of methods**. … .

The program code utilized within a multi-tiered architecture therefore is not localized "monolithic" code. Rather, it is composed of many packages or a "plurality of archive files" as presently recited by Applicants, in which each of the archive files have respective classes or class files, and each such class or class file sets forth "methods," which, of course, provide the underlying functionality.

Even more problematically, these packages or "archive files" having the underlying functionality must be **dispersed across the various tiers of a multi-tier architecture** to provide for the beneficial scalability yielded by the multi-tier architectural model. Again, refer to Applicants' specification which teaches:

> [**0020**] The multi-tiered architecture illustrated in Figure 2b **may be implemented using a variety of different object-oriented application technologies** <u>**at *each of the layers* of the multi-tiered architecture**</u>, including those based on the Java 2 Enterprise EditionTM ("J2EE").

Because the code must be implemented via functionality, packages, or archive files "at **each of the layers** of [a] multi-tiered architecture," the corresponding functionality, packages, or archive files, if they are to be traced, must be selected from among the disparate layers of the multi-tiered architecture.

Hence, Applicants expressly recite within independent claim 40 as amended herein:

> modifying bytecode of **only a subset** of a plurality methods from which the application is composed … **wherein the two or more archive files are associated with <u>two or more distinct tiers</u> of the multi-tiered architecture** …

Thus, the "subset" of methods which are operated upon are not simply from a monolithic code base, such as that from the conventional arts, nor are subset of methods merely from multiple archive files, but rather, the methods are expressly recited as being "selected from two or more class files in **two or more archive files** composing the application's bytecode … wherein the two or more archive files are **associated with two or more distinct tiers of the multi-tiered architecture**."

Applicants note that this particular limitation was not present in the proposed claims discussed with Examiner Rutten, but is introduced herein as additional clarifying detail and distinction over the presently relied upon references cited by the Office Action.

Cited references fail to disclose at least one limitation:

Turning now to Applicants' expressly recited limitations, Applicants respectfully submit that Berry fails to disclose at least one limitation which Applicants recite within the amended claims. For example:

1) Berry fails to disclose selection of disparate packages from among a multi-tiered architecture. In amended independent claim 40, Applicants expressly recite:

> **modifying bytecode of only a subset of a plurality methods from which the application is composed**, the subset of the plurality of methods selected from two or more class files in two or more archive files composing the application's bytecode as specified via the Graphical User Interface presented to the end-user device, **wherein the two or more archive files are associated with *two or more distinct tiers* of the multi-tiered architecture**, and wherein the modified subset of the plurality of methods specified provides the user-configurable level of granularity by providing the functionality for tracing the program flow of the application through only the subset of the plurality of methods specified via the tracing and debugging operations injected into the subset of the plurality of methods specified;

Although the Office Action relies upon Berry in its rejection of Applicants' claims, Berry makes no disclosure whatsoever of a multi-tier architecture. Accordingly, Berry does not, and indeed cannot, support an interpretation that "bytecode of only a subset of a plurality of methods" are modified, "wherein **the two or more archive files are associated with two or more *distinct tiers* of a multi-tiered architecture**, such as that which Applicants recite in amended claim 40.

2) Berry fails to disclose Applicants' chosen **mechanism by which less than all methods** of an application to be modified are selected. In the present Office Action at page 3, first paragraph, the Examiner points out that Berry provides "**implicit support** for modification" of less than all methods, specifically stating:

> the inclusion/exclusion lists provide for the modification of **all** methods, the modification of **none** of the methods, **and everything in-between**.

Applicants must strongly disagree with such an interpretation. Turning to Berry specifically, the relevant passage at column 7, lines 5-8 discloses:

> Selective instrumentation is possible if only some of the methods are to be instrumented. **In the described embodiment, an *inclusion/exclusion list* is used to specify which methods are to be instrumented**. However, any number of procedures may be used to determine whether a particular method is to be instrumented or to specify which methods are to be instrumented.

Thus, Berry does disclose that an "inclusion/exclusion list" may be used "to specify which methods are to be instrumented." However, Berry provides nothing more regarding the "inclusion/exclusion" lists. Such a feature is not elaborated upon. Such a feature is not depicted in the Figures of Berry. Such a feature is not discussed elsewhere by Berry.

It is unknown from the reference, and indeed unknowable, what mechanism Berry uses to include or exclude methods using the list given that Berry makes no disclosure whatsoever

regarding **how** such a list is established, populated, referenced, or implemented. There is simply nothing more provided by Berry than the fact that an "inclusion/exclusion list" may be used.

Conversely, Applicants recite far more detail within the body of independent claim 40 as amended herein. For example, Applicants recite:

> **modifying bytecode of only a subset** of a plurality methods from which the application is composed **... as specified via the Graphical User Interface presented to the end-user device.**

Thus, Berry specifically fails to disclose that a selection is "specified via [a] Graphical User Interface presented to the end-user device" as claimed by Applicants.

The Examiner relies upon an implicit disclosure, however, the requirements for relying upon a reference as disclosing something which is not expressly stated are very strict, as the Examiner is certainly aware. For example, to rely upon a reference as disclosing that which is not expressly stated, it must be shown that a particular limitation **necessarily flows** from that which is actually disclosed. Refer to M.P.E.P. § 2112.

Berry might use lists that are hard-coded into the program relying upon the lists, a mechanism which is consistent with that which is actually disclosed by Berry, but is insufficient to anticipate Applicants express limitation. We do not know. Berry does not say. We can only guess what Berry might use as a mechanism, given that Berry is silent with respect to such details. Accordingly, it is improper to presume that Berry discloses that which is claimed by Applicants, as doing so is impermissible conjecture.

Further still, Berry does not disclose that the "inclusion/exclusion list [] used to specify which methods" are selected from an "application [] composed of a plurality of archive files **associated with two or more distinct tiers in a multi-tiered architecture**," given that Berry provides no detail whatsoever as to **how** the "inclusion/exclusion" list is created or implemented.

Nor does Berry provide any discussion of a "multi-tiered architecture," much less how to select a subset of methods from such an architecture.

Because Berry fails to disclose at least one limitation which Applicants recite in independent claim 40, Applicants respectfully submit that claim 40 is patentable over Berry.

At page 4, last paragraph, the Office Action correctly concedes that Berry fails to disclose particular limitations which Applicants recite in independent claim 40, but nevertheless asserts that Ims cures the admitted deficiencies of Berry.

The Ims reference, however, whether considered individually or in combination with Berry, does not cure the deficiencies of Berry as discussed above with respect to independent claim 40 because Ims similarly is silent with respect to:

> **modifying bytecode of only a subset of a plurality methods** ... in two or more archive files ... **wherein the two or more archive files are associated with *two or more distinct tiers* of the multi-tiered architecture**

such as that which Applicants expressly recite in independent claim 40 as amended herein. Ims is further similarly silent with respect to:

> **modifying bytecode of only a subset** of a plurality methods from which the application is composed **... as specified via the Graphical User Interface presented to the end-user device**

as Applicants recite in independent claim 40 as amended herein.

Because the combination of Berry and Ims fails to disclose at least one limitation as Applicants recite in independent claim 40, as amended herein, Applicants respectfully submit that independent claim 40 is patentable over the references and in condition for allowance. Applicants further submit that independent claims 45 and 50, which recite similar limitations to those of independent claim 40 specifically discussed above as missing from the combination of

Berry and Ims, as well as those claims which depend directly or indirectly upon independent claims 40, 45 and 50, and thus incorporate the limitations of their respective parent claims, are also patentable over the references and in condition for allowance for at least the same reasons as stated above with respect to independent claim 40 rejected under 35 U.S.C. § 103.

Accordingly, Applicants respectfully request the Examiner to withdraw the rejection to the claims under 35 U.S.C. §103.

CONCLUSION

Given the above remarks, all claims pending in the application are in condition for allowance. If the undersigned attorney has overlooked subject matter in any of the cited references that is relevant to allowance of the claims, the Examiner is requested to specifically point out where such subject matter may be found. Further, if there are any informalities or questions that can be addressed via telephone, the Examiner is encouraged to contact the undersigned attorney at (503) 439-8778.

Charge Deposit Account

Please charge our Deposit Account No. 02-2666 for any additional fee(s) that may be due in this matter, and please credit the same deposit account for any overpayment.

Respectfully Submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

/Gregory D. Caldwell/	May 17, 2010
Gregory D. Caldwell	Date
Registration No. 39,926	
Attorney for Applicants	

Blakely, Sokoloff, Taylor & Zafman LLP
1279 Oakmead Parkway
Sunnyvale, CA 94085-4040
Telephone: (503) 439-8778
Facsimile: (503) 439-6073